
Subject: [PATCH 07/25] sysfs: Rewrite sysfs_drop_dentry.
Posted by [ebiederm](#) on Tue, 07 Aug 2007 21:14:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Currently we find the dentry to drop by looking at sd->s_dentry.
We can just as easily accomplish the same task by looking up the
sysfs inode and finding all of the dentries from there, with the
added bonus that we don't need to play with the sysfs_assoc_lock.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

fs/sysfs/dir.c | 53 ++++++-----
1 files changed, 26 insertions(+), 27 deletions(-)

diff --git a/fs/sysfs/dir.c b/fs/sysfs/dir.c

index ef99883..6ca4382 100644

--- a/fs/sysfs/dir.c

+++ b/fs/sysfs/dir.c

@@ -563,50 +563,49 @@ void sysfs_remove_one(struct sysfs_addrm_cxt *acxt, struct
sysfs_dirent *sd)

* Drop dentry for @sd. @sd must have been unlinked from its
 * parent on entry to this function such that it can't be looked
 * up anymore.

- *

- * @sd->s_dentry which is protected with sysfs_assoc_lock points
- * to the currently associated dentry but we're not holding a
- * reference to it and racing with dput(). Grab dcache_lock and
- * verify dentry before dropping it. If @sd->s_dentry is NULL or
- * dput() beats us, no need to bother.

*/

static void sysfs_drop_dentry(struct sysfs_dirent *sd)

{

- struct dentry *dentry = NULL;

struct inode *inode;

+ struct dentry *dentry;

+

+ inode = ilookup(sysfs_sb, sd->s_ino);

+ if (!inode)

+ return;

- /* We're not holding a reference to ->s_dentry dentry but the
- * field will stay valid as long as sysfs_assoc_lock is held.

+ /* Drop any existing dentries associated with sd.

+ *

+ * For the dentry to be properly freed we need to grab a
+ * reference to the dentry under the dcache lock, unhash it,
+ * and then put it. The playing with the dentry count allows
+ * dput to immediately free the dentry if it is not in use.

```

    */
- spin_lock(&sysfs_assoc_lock);
+repeat:
    spin_lock(&dcache_lock);
-
- /* drop dentry if it's there and dput() didn't kill it yet */
- if (sd->s_dentry && sd->s_dentry->d_inode) {
-     dentry = dget_locked(sd->s_dentry);
+ list_for_each_entry(dentry, &inode->i_dentry, d_alias) {
+     if (d_unhashed(dentry))
+         continue;
+     dget_locked(dentry);
        spin_lock(&dentry->d_lock);
        __d_drop(dentry);
        spin_unlock(&dentry->d_lock);
+     spin_unlock(&dcache_lock);
+     dput(dentry);
+     goto repeat;
    }
-
    spin_unlock(&dcache_lock);
- spin_unlock(&sysfs_assoc_lock);
-
- dput(dentry);

    /* adjust nlink and update timestamp */
- inode = ilookup(sysfs_sb, sd->s_ino);
- if (inode) {
-     mutex_lock(&inode->i_mutex);
+ mutex_lock(&inode->i_mutex);

-     inode->i_ctime = CURRENT_TIME;
+ inode->i_ctime = CURRENT_TIME;
+ drop_nlink(inode);
+ if (sysfs_type(sd) == SYSFS_DIR)
+     drop_nlink(inode);
-     if (sysfs_type(sd) == SYSFS_DIR)
-         drop_nlink(inode);

-     mutex_unlock(&inode->i_mutex);
-     iput(inode);
- }
+ mutex_unlock(&inode->i_mutex);
+
+ iput(inode);
    }

    /**

```

--

1.5.1.1.181.g2de0

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
